On the existence of orthogonal decompositions of the simple Lie algebra of type C_3

A.Torstensson

December 16, 2005

1 Orthogonal decompositions of simple Lie algebras

The basic question, of which we will study a special case, is whether all classical simple Lie algebras can be decomposed into a direct sum of Cartan subalgebras which are orthogonal to each other in the sense that the Killing form $B(X,Y) = 0$ if X and Y are elements from different Cartan subalgebras occurring in the decomposition. This problem has been studied for all four infinite series of classical simple Lie algebras, and OD:s have been constructed in all cases except for A_n when $n+1$ is not a prime power and C_n when n is not a power of 2 [?]. In all other cases the existence of an OD is an open question. One result pointing to a negative answer to this question is that A_5 has no OD of monomial type [?]. The main result of this paper is that C_3 has no monomial orthogonal decomposition either.

2 Description of the problem in the C_3 situation

Let L be the Lie algebra of type C_n over $\mathbb{C}$. It can be realized as the $2n \times 2n$ matrices of the form $\begin{pmatrix} A & B \\ C & -A^t \end{pmatrix}$ where A, B and C are $n \times n$ matrices and B and C are symmetric. The multiplication in L is the usual bracket multiplication $[X,Y] = XY - YX$.

In a semisimple Lie algebra a Cartan subalgebra can be characterized as a maximal toral subalgebra. Taking into consideration that a toral algebra is always abelian we can also think of Cartan subalgebras as the maximal ones among the abelian subalgebras generated by semisimple elements. [**?**].

According to the well known Cartan-Chevelley theorem [**?**] any two Cartan subalgebras in a finite dimensional Lie algebra over an algebraically closed field are conjugate under some automorphism of the Lie algebra. Two consequences of this is that all Cartan subalgebras have the same dimension and that the existence of an OD implies the existence of an OD containing the standard Cartan subalgebra, H, consisting of all the diagonal matrices in C_3.

We now turn to the notion of orthogonality. The Killing form $B(X,Y) = trace(adXadY)$ is a symmetric, invariant bilinear form on any Lie algebra and in charcteristic zero it is nondegeneration is a criterion for semisimplicity. [**?**].

In a simple Lie algebra, L, over an algebraically closed field there is only one such form up to multiplication with a constant from the underlying field. This can be seen in the following way. Let f and g be two nondegenerate, invariant, bilinear forms on L. For fixed $X \in L$ let $S(Y) = g(X,Y)$. Then since f is nondegenerate we must have $S(Y) = f(Z,Y)$ for some $Z \in L$. We define the mapping θ by $\theta(X) = Z$. Then θ is a linear mapping from L to L and it follows from the invariance of g that it commutes with adL. Hence, by Schurs lemma, θ is some multiple of the identity.

Especially we have $B(X,Y) = c \cdot trace(XY)$ on C_3 where c is a nonzero complex number. This means that X and Y are orthogonal if and only if $trace(XY) = 0$.

Also note that from the fact that the Killing form is nondegenerate on any Cartan subalgebra (see for example [**?**]) it follows that a sum of orthogonal Cartan subalgebras always is direct.

Thus, in concrete terms, the problem of finding an orthogonal decomposition of C_3 is to find out if there exists 7 sets each containing 3 commuting, linearly independent, diagonalizable matrices in C_3 such that any matrices X, Y from different sets satisfy $trace(XY) = 0$. The simplest possible type of OD would be a partition of the basis matrices obtained from the root space decomposition with respect to the standard Cartan subalgebra H. This turned out to be a successful idea for the Lie algebras of types B_n and D_n [**?**]. In our case, however, we need a partition into 7 sets with commuting elements such that basis matrices from different sets are orthogonal. This is clearly impossible, since there are nine pairs of non-orthogonal basis matrices and from dimensional considerations we can see that matrices from two different pairs can never be in the same set. This forces us into having at least nine sets.

We can assume that one of these sets is a basis for H, for example $E_1 = diag(1,0,0,-1,0,0), E_2 = diag(0,1,0,0,-1,0), E_3 = diag(0,0,1,0,0,-1)$ A matrix in C_3 is orthogonal to all three basis matrices, i. e. orthogonal to H, if and only if it has zeroes on the diagonal. This is immediate since the trace of the product of a matrix of type 2 and E_i is two times the i:th diagonal element of the submatrix A.

3 Monomial decompositions

A nonsingular matrix is called monomial if it can be written as a product of a diagonal matrix and a permutation matrix. Consequently the diagonal matrix has no zeroes on the diagonal. An orthogonal decomposition is called monomial if there exists a basis consisting of monomial matrices for each Cartan subalgebra that occurs.

The known constructions of OD:s for Lie algebras of type A_n and C_n produce monomial decompositions so we will concentrate on the existence of such an OD of C_3. This means that we can choose basis matrices in $H_1, H_2, ..., H_6$ that are monomial, diagonalizable and have no zeroes on the diagonal. The diagonlizability actually follows from the fact that the matrix is monomial:

Theorem 3.1 *A monomial matrix DP is diagonalizable. If $D = diag(d_1, d_2, ..., d_n)$ and $(i\ P(i)\ \cdots\ P^m(i))$ occurs in the decomposition of P into disjoint cycles then all $m+1$ roots of $\lambda^{m+1} = d_i d_{P(i)} \cdots d_{P^m}$ are eigenvalues of DP.*

Proof: We show that DP is diagonalizable explicitly by constructing n linearly independent eigenvectors. We look at each cycle $(i\ P(i)\ \cdots\ P^m(i))$ (where we regard P as a permutation) separately. Let λ satisfy $\lambda^{m+1} = d_i d_{P(i)} ... d_{P^m(i)}$. Since all the d_i:s are nonzero we get an eigenvector, $v = (v_1, v_2, \ldots, v_n)$ by putting $v_i = 1, v_{P(i)} = \frac{\lambda}{d_i}, ..., v_{P^m(i)} = \frac{\lambda^m}{d_i d_{P(i)} ... d_{P^{m-1}(i)}}$ and all other v_k:s equal to zero. It is easy to see that all these vectors are linearly independent so they constitute a basis of eigenvectors so the matrix is diagonalizable. ♦

4 Possible monomial basis matrices

Our first goal is to find all permutations P such that DP may be one of the matrices of a monomial basis for some Cartan subalgebra orthogonal to H. It follows from orthogonality that the permutation has no fixed point since all diagonal elements must be zero. Also, since every H_i consists of the matrices of the form $A \begin{pmatrix} D & 0 \\ 0 & -D \end{pmatrix} A^{-1}$ for some automorphism $X \mapsto AXA^{-1}$ of C_3 we must have that Y^3 is in H_i if Y is in H_i, which excludes those matrices containing a three cycle.

It remains to consider all permutation matrices, P, of the form

$$\begin{pmatrix} A & B \\ C & A^t \end{pmatrix}$$

with B and C symmetric and $trace(P) = trace(P^3) = 0$. Let us calculate the number of such matrices. We consider different cases depending on the total number of ones in the submatrix A, which we denote by $|A|$.

If $|A| = 0$ we can choose B and C as any symmetric permutation matrix of three elements. Because of the symmetry we must have one or three elements on the diagonal, i e one or three fixed points. This gives four different permutations resulting in $4 * 4 = 16$ possible P. (Both trace conditions are automatically satisfied here.)

The next case is $|A| = 1$. There are 6 such matrices A with no diagonal entries. For each fixed A there are four ways of choosing B and C and exactly one of them contains a three cycle: Let (i, j) be the nonzero entry of A. It follows that $(j + 3, i + 3)$ also is nonzero and all other entries in the right lower block of P are zero. Then P is a permutation matrix if and only if P with rows i and $j + 3$ and columns j and $i + 3$ removed is a permutation matrix of 4 elements. The new matrix, let us call it P', has the structure

$$\begin{pmatrix} 0 & B' \\ C' & 0 \end{pmatrix}$$

where B' is B with row i and column i removed and C' is C with row j and column j removed. In this situation the symmetry of B and C is equivalent to that of B' and C'. This gives us exactly two possibilities for each of them, that is 4 possible combinations. When do we get a three cycle? This happens if and only if i is in a three cycle (since we have no fixed points). Let us call this cycle (ijk). In how many ways can we choose k? The conditions are that $k = 4, 5, 6$ and $k \neq i + 3$ and $k \neq j + 3$. Since $i \neq j$ there is exactly one such k. Thus, we have 6 cases with $|A| = 1$ each giving 3 matrices satisfying all conditions.

There are 9 ways to arrange two ones in A and each case determines P uniquely since the matrices B' and C' that we get after removing the already determined rows and columns are of size one. It is easy to see that B' and C' are originally diagonal elements of B and C so that P is of the right form. We only have to check that we cannot have any three cycle. If the nonzero entries of A are (i, j) and (k, l) we consider two cases. If $j = k$ then the only possible three cycle containing i is (ijl) but since both l and i are between 1 and 3 (l, i) must be equal to (i, j) or (k, l). In either case the three cycle degenerates into a transposition. If $j \neq k$ the third element in (ijs) must be between 4 and 6. It follows that $i = P(s) = s - 3$ which leads to the contradiction $j + 3 = P^{-1}(i + 3) = P^{-1}(s) = j$. Thus, all our 9 permutations are of the desired form.

If $|A| = 3$ there is no way to avoid a three cycle, since A must give a permutation of $1, 2, 3$ without any fixed point.

Altogether we have 43 matrices satisfying the three conditions we have stated and they can easily be constructed using the argumentation above. A list of the permutations is given in the appendix.

5 Possible monomial Cartan subalgebras

In this section we investigate in which ways we can combine the permutations from the previous section so that they are the permutation part of the basis matrices for a Cartan subalgebra orthogonal to H. In other words we are looking for all maximal abelian subalgebras containing three linearly independent monomial matrices. One method is to start with some monomial matrix and find which other such matrices commutes with it. The simplest case is when all eigenvalues are different.

Proposition 1 *Let M be an element in C_3 with only simple eigenvalues. Then there is only one Cartan subalgebra containing M and it consists of all odd polynomials in M.*

Proof: Let P denote the subalgebra of C_3 consisting of matrices that are polynomials in M, and C some Cartan subalgebra containing M.

$C \subseteq P$: Let B be an element in the Cartan subalgebra. Since M and B commute we can choose a basis in which they are both diagonal. It follows from the simplicity of M:s eigenvalues that I,M, M^2, M^3, ... M^5 are linearly independent (Vandermonde's determinant). Especially M, M^3 and M^5 are linearly independent diagonal elements of C_3 and hence any diagonal matrix in C_3, especially B, is a linear combination of them.

$P \subseteq C$: Let D be an element in P. Then all elements in C must commute with D, which makes $C \bigoplus \lambda D$ a commutative subalgebra of C_3. It follows from the maximality of C as abelian subalgebra of C_3 that $D \in C$.

The polynomials in M that are in C_3 are exactly those containing only odd powers which can be seen in the following way. M, being diagonalizable, is inside some Cartan subalgebra since it can be extended to a maximal abelian subalgebra consisting of diagonalizable elements. There is some C_3-automorphism $A \mapsto X^{-1}AX$ diagonalizing M since all Cartan subalgebras are conjugate. Then $X^{-1}MX = diag(\lambda_1, \lambda_2, \lambda_3, -\lambda_1, -\lambda_2, -\lambda_3)$. Now, if p is any polynomial,

$$p(M) = Xp(X^{-1}MX)X^{-1} =$$

$$= X diag(p(\lambda_1), p(\lambda_2), p(\lambda_3), p(-\lambda_1), p(-\lambda_2), p(-\lambda_3))X^{-1}$$

which is in C_3 if and only if $diag(p(\lambda_1), p(\lambda_2), p(\lambda_3), p(-\lambda_1), p(-\lambda_2), p(-\lambda_3))$ is in C_3. It is clear that the last matrix is in C_3 exactly when p only contains odd powers. ♦

It follows from theorem 3.1 that the theorem above is applicable whenever P is a six cycle. This gives all the subalgebras of the kind we are looking for containing a six cycle in some monomial basis. For the rest of the section we will assume that none of the basis matrices is a six cycle.

Let us look at the product of two monomial matrices DP and $D'P'$. The entry (i, j) in the product is 0 if $P'(P(i)) \neq j$ and $d_i d'_{P(i)}$ otherwise. The two matrices commute if and only if $d_i d'_{P(i)} \delta_{P'(P(i)),j} = d'_i d_{P'(i)} \delta_{P(P'(i)),j}$ for all i, j. Now, since all diagonal elements are nonzero, this implies that the two permutations P and P' commute. A first step is therefore to find out when two permutations of the kind we are interested in commute.

Proposition 2 *a) The permutations commuting with a six cycle, P, are all powers of P.*

b) The permutations commuting with $(a_1a_2)(a_3a_4a_5a_6)$ are all products ST where $S \in \{id, (a_1a_2)\}$ and T is a power of $(a_3a_4a_5a_6)$.

c) The permutations commuting with $(a_1a_2)(a_3a_4)(a_5a_6)$ are all products ST where $S \in \{id, (a_1a_2), (a_3a_4), (a_5a_6), (a_1a_2)(a_3a_4), (a_1a_2)(a_5a_6), (a_3a_4)(a_5a_6), (a_1a_2)(a_3a_4)(a_5a_6)\}$ and $T \in \{(a_1a_3)(a_2a_4), (a_1a_5)(a_2a_6), (a_3a_5)(a_4a_6)\}$.

Proof: If we let S_6 act on a given permutation with conjugation the stabilizer consists of the permutations commuting with the given one and the orbit consist of all permutations which have the same structure (i e if we write the two permutations P and P' as products of disjoint cycles there is a one-to-one correspondence between the cycles in P and P' such that corresponding cycles have equal length). We know that $|stabilizer| = |group|/|orbit|$. Hence, to find the number of commuting permutations we only need to calculate the length of the corresponding orbit. There are 5! six cycles in S_6 so a cycle has $6!/5! = 6$ commuting elements. The length of the orbit in case (b) is $\binom{6}{2} 3!$ since after choosing the elements to be in the transposition we can order the remaining elements in 3! ways. A similar argument in case (c) show that the orbit has length 30. From this it follows that the number of permutations stated in the theorem is correct. It is obvious that they do commute and straightforward to check that they are all different. ♦

Let us first exhibit the simplest case. What if we use the same permutation more than once in the basis. Then the permutations certainly commute but can we make the corresponding monomial matrices both commuting and linearly independent?

Before we can state the next proposition we will need the concept of related cycles. Two cycles are related if, for some i between 1 and 3, one of them contains i and the other $i+3$. This might not seem like a natural relation, but the proof below will clarify its importance.

Proposition 3 *The maximum number of linearly independent commuting monomial matrices in C_3 with the same permutation P is equal to the maximal number of unrelated cycles in P.*

Proof: The condition of commutativity in this situation is that the quotients d_i/d_i' (d_i and d_i' being the entries of the two diagonal matrices) are equal for all i in the same cycle of P. When DP is in C_3 we also have some additional conditions on the d_i, namely

A. $(DP)_{i+3,j+3} = -(DP)_{j,i}$, $1 \leq i,j \leq 3$ B. $(DP)_{i,j+3} = (DP)_{j,i+3}$, $1 \leq i,j \leq 3$ C. $(DP)_{i+3,j} = (DP)_{j+3,i}$, $1 \leq i,j \leq 3$

Let us take a closer look at condition A. Assume that we can find i and j such that $P(j) = i$. (Otherwise A is an empty condition.) Then A says that $d_{i+3} = -d_j$ and $d'_{i+3} = -d'_j$. If we have two matrices DP and $D'P$ of this kind two cases can occur. If $i+3$ and j (or equivalently i and $i+3$) are in the same cycle the conditions of commutativity are not affected but if they are in different cycles the quotients d_i/d'_i must be equal for the two cycles. Considering B and C in the same way we can formulate the conditions for finding equivalent cycles as follows:

A. $P(i) = j$ with j and $j+3$ in different cycles B. $P(i) = j+3$ with j and $j+3$ in different cycles C. $P(i+3) = j$ with j and $j+3$ in different cycles

It is easy to see that if we fix j between 1 and 3 we either have j and $j+3$ in the same cycle, in which case none of the conditions is satisfied or we have j and $j+3$ in different cycles and then exactly one of the conditions A and C are satisfied. This means that the cycles must have equal quotient if and only if they are related in the sense defined above.

Given a number of matrices DP, $D'P$, $D''P$, ... that commute with each other we want to find the maximal number of linearly independent such matrices. This is equivalent to the vectors $(d_1, d_2, ..., d_6)$ being linearly independent and by permuting indices we may assume that all indices belonging to related cycles are consecutive. Let m be the maximal number of unrelated cycles and $d_{(1)}$, $d_{(2)}$, ... ,$d_{(m)}$ the subvectors of $(d_1, d_2, ..., d_6)$ corresponding to the different classes of unrelated cycles. Then $d'_{(i)} = \alpha'_i d_{(i)}$, $d''_{(i)} = \alpha''_i d_{(i)}$ and so on. Look at the matrix having the vectors $(d_1, d_2, ..., d_6)$ as rows. The number of linearly independent columns is obviously m, the number of unrelated cycles. The statement follows since the rowrank and columnrank of a matrix coincide. ♦

To complete our investigation we must find all commuting pairs of monomial matrices with different permutations. The six cycle case is already covered so we may look only at pairs of other permutations. Seven of the permutations corresponding to the 43 matrices from the previous section are products of three transpositions. They are given by

$$(12)(36)(45), (13)(25)(46), (14)(23)(56), (14)(25)(36), (14)(26)(35),$$

$$(15)(24)(36), (16)(25)(34).$$

There are 12 permutations that are products of a four cycle and a transposition: $(1254)(36), (1364)(25)$, $(2365)(14), (2536)(14), (1425)(36), (1436)(25)$ and third powers of these.

Using proposition 2 we can find the commuting pairs a permutations. In each case we create the corresponding monomial matrices and solve for the diagonal elements under the condition of commutativity. One finds that $P = (14)(25)(36)$ plays a special role. It commutes with all monomial matrices of the first type. Any other monomial matrix of the first type commutes only with P and itself. A monomial matrix DP where P is of the second kind commutes only with monomial matrices made from P or P^3.

We can now describe all the possible choices of permutations P_1, P_2, P_3 such that D_1P_1, D_2P_2, D_3P_3 is a Cartan subalgebra for some suitable choice of diagonal matrices.

Proposition 4 *There are at most 37 ways to choose the three permutations P_1, P_2, P_3 such that there exists nondegenerate diagonal matrices D_1, D_2, D_3 that makes $< D_1P_1, D_2P_2, D_3P_3 >$ a Cartan subalgebra.*

Proof: If one of the permutations, P, is a six cycle it follows from proposition 1 that there is a unique way to choose the permutations in the basis. It also follows that P and P^5 give the same choice of permutations so our 24 six cycles gives us 12 types of Cartan subalgebras.

The remaining case is triples of permutations where none is a six cycle. Using proposition 3 one can see by inspection that we can have at most two occurrences of the same permutation in a basis except for $(14)(25)(36)$ of which we can have three. From our investigation of commutativity above it follows that the possibilities are the following: Each of the permutations $(12)(36)(45), (13)(25)(46), (14)(23)(56), (14)(26)(35), (15)(24)(36), (16)(25)(34)$ can occur either once or twice in the basis and can only be combined with $(14)(25)(36)$. This gives us 13 subalgebras. For each P of the four cycle type we have two possible triples: (P, P, P^3) and (P, P^3, P^3) resulting in 12 additional subalgebras. Altogether this gives 37 sets of permutations. ♦

Remark: Even though we will not use this result it is interesting to note that computer calculations show that for each triple it is possible to construct D_1, D_2, D_3 such that D_1P_1, D_2P_2, D_3P_3 commutes. In other words we can replace "at most" with "exactly" in the proposition. Let us look at an example of such a calculation. If we look at the subalgebra where $P = (12)(36)(45)$ occurs twice and $Q = (14)(25)(36)$ once and let the basis matrices be $diag(d_1, d_2, d_3, d_4, d_5, d_6) * P$, $diag(e_1, e_2, e_3, e_4, e_5, e_6) * P$ and $diag(f_1, f_2, f_3, f_4, f_5, f_6) * Q$ the conditions for being in C_3 are $d_4 = -d_2, d_5 = -d_1, e_4 = -e_2, e_5 = -e_1$ and those for commutativity are $d_1e_2 = d_2e_1, d_3e_6 =$

$d_6e_3, d_4e_5 = d_5e_4, d_1f_2 = f_1d_4, d_2f_1 = f_2d_5, d_3f_6 = d_6f_3, d_4f_5 = f_4d_1, d_5f_4 = f_5d_2, e_1f_2 = f_1e_4, e_2f_1 = f_2e_5, e_3f_6 = e_6f_3, e_4f_5 = f_4e_1, e_5f_4 = f_5e_2$. The system has the solutions $d_1 = -d_5, d_4 = -\frac{d_5f_4}{f_5}, e_1 = -e_5, d_3 = \frac{d_6f_3}{f_6}, e3 = \frac{e_6f_3}{f_6}, e_4 = -\frac{e_5f_4}{f_5}, f_1 = \frac{f_2f_5}{f_4}, e2 = \frac{e_5f_4}{f_5}, d_2 = \frac{d_5f_4}{f5}$

A list of the 37 triples of permutations is included in the appendix.

6 The impossibility of orthogonality

For these 37 kinds of monomial Cartan subalgebras we would like to find out which can be made orthogonal to each other. If we have two such subalgebras $< D_1P_1, D_2P_2, D_3P_3 >$ and $< D'_1P'_1, D'_2P'_2, D'_3P'_3 >$ they cannot be orthogonal if some of the permutations $P_iP'_j$ has exactly one fixed point. It turns out that otherwise the two subalgebras are orthogonal if we choose the D_i and the D'_i in a suitable way except for 12 exceptional pairs stated in the appendix. It is interesting to note that each six cycle subalgebra appears in exactly two exceptional pairs. The adjacency matrix representing orthogonality of subalgebras can also be found in the appendix. Unfortunately the graph contains complete subgraphs of order 12 and 13 which shows that we have more than 30940 six tuples of subalgebras which are pairwise orthogonal. The time it takes to check if six subalgebras with given permutations can give an orthogonal decomposition is approximately 20 minutes (using Maple V). The total time with this approach would be at least one year. To reduce the amount of calculations required to examine all potential decompositions we look instead at triples of orthogonal subalgebras. Solving such a system takes only about one minute and there are 2091 triples which are pairwise orthogonal. In the appendix is a complete list of all orthogonal triples.

Using this data we can exclude some subalgebras. Assume that $H1$ is in the decomposition. Then some of the seven triples containing $H1$ must be in the decomposition. Look at the first case: $(H1, H15, H28)$ is in the decomposition. If H is another subalgebra in the decomposition then $(H1, H15, H)$ is a triple so $H = H28$ or $H = H29$. On the other hand $(H1, H28, H)$ is a triple which gives that H is $H15$, $H16$ or $H19$, a contradiction. This kind of argument excludes all decompositions with a six cycle subalgebra except those containing 4 copies of $H19$, but since $H19$ contains only one permutation this contradicts the linear independence of all basis matrices.

Using a computer we were able to find all sets of six subalgebras such that any three of them is one of the orthogonal triples above. Since the subalgebra $H19$ occurred much more frequently than the other subalgebras it was useful to add the condition that no more than two such subalgebras can be in the same decomposition. This resulted in only 27 six tuples listed in the appendix.

For each of them we solved the system of equations that comes from the conditions of commutativity and orthogonality and in no case this system had any solutions. This completes the proof of our main theorem.

Theorem 6.1 *C_3 has no monomial orthogonal decomposition containing H.*

A Data and Maple scripts

A.1 The possible basis permutations

P1=(123654), P2=(125463), P3=(125436), P4=(126354), P5=(136452), P6=(132564), P7=(136425), P8=(135264), P9=(145632), P10=(145362), P11=(146523), P12=(142365), P13=(146253), P14=(143256), P15=(142635), P16=(143526), P17=(152463), P18=(156324), P19=(153624), P20=(152436), P21=(163452), P22=(165234), P23=(162534), P24=(163425), P25=(12)(36)(45), P26=(13)(25)(46), P27=(14)(23)(56), P28=(14)(25)(36), P29=(14)(26)(35), P30=(15)(24)(36), P31=(16)(25)(34), P32=(1254)(36), P33=(1364)(25),P34=(1452)(36), P35=(14)(2365), P36=(1463)(25), P37=(14)(2563), P38=(14)(2536), P39=(1425)(36), P40=(1436)(25), P41=(14)(2635), P42=(1524)(36), P43=(1634)(25)

A.2 The possible Cartan subalgebras

In this list (i,j,k) refers to the triple (Pi,Pj,Pk) of permutations from the above list.

Six cycle type: H1=(1,31,9), H2=(2,29,5), H3=(3,27,21), H4=(4,26,10), H5=(6,30,11), H6=(7,27,17), H7=(8,25,13), H8=(12,26,18), H9=(14,25,22), H10=(15,31,19), H11=(16,30,23), H12=(20,29,24)

Three transposition type: H13=(25,25,28), H14=(25,28,28), H15=(26,26,28), H16=(26,28,28), H17=(27,27,28), H18=(27,28,28), H19=(28,28,28), H20=(29,29,28), H21=(29,28,28), H22=(30,30,28), H23=(30,28,28), H24=(31,31,28), H25=(31,28,28)

One transposition type: H26=(32,32,34), H27=(32,34,34), H28=(33,33,36), H29=(33,36,36), H30=(35,35,37), H31=(35,37,37), H32=(38,38,41), H33=(38,41,41), H34=(39,39,42), H35=(39,42,42), H36=(40,40,43), H37=(40,43,43)

The following Maple script was used for checking that a Cartan subalgebra can be constructed for each triple above:

```
cartantest := proc(k, l, m)
  local Q, eqs, j, L, L1, L2, L3, L4, S1, S2, S3, r, s, H1, H2, H3;
    Q[1] := P.k ;
    Q[2] := P.l ;
    Q[3] := P.m ;
    eqs := {} ;
    for j to 3 do
      L[j] := evalm(diag(d1.j, d2.j, d3.j, d4.j, d5.j, d6.j) `&*` Q[j]) ;
      print(L[j]) ;
      L1[j] := submatrix(L[j], 1..3, 1..3) ;
      L2[j] := submatrix(L[j], 1..3, 4..6) ;
      L3[j] := submatrix(L[j], 4..6, 1..3) ;
      L4[j] := submatrix(L[j], 4..6, 4..6) ;
      S1[j] := evalm(L1[j] + transpose(L4[j])) ;
      S2[j] := evalm(L2[j] - transpose(L2[j])) ;
      S3[j] := evalm(L3[j] - transpose(L3[j])) ;
      eqs := eqs union {d5.j <> 0, d6.j <> 0, d1.j <> 0, d2.j <> 0, d3.j <> 0, d4.j <> 0} ;
      for r to 3 do
        for s to 3 do eqs := eqs union {S2[j][r, s] = 0, S3[j][r, s] = 0, S1[j][r, s] = 0} od
      od
    od;
    H1 := evalm((L[1] `&*` L[2]) - (L[2] `&*` L[1])) ;
    H2 := evalm((L[1] `&*` L[3]) - (L[3] `&*` L[1])) ;
    H3 := evalm((L[2] `&*` L[3]) - (L[3] `&*` L[2])) ;
    for r to 6 do
      for s to 6 do eqs := eqs union {H3[r, s] = 0, H2[r, s] = 0, H1[r, s] = 0} od
    od;
    print(eqs) ;
    RETURN(solve(eqs))
  end
```

A.3 The orthogonal pairs

There exist orthogonal subalgebras of types Hi and Hj if and only if the entry (i, j) in the matrix below is one.

$$
\begin{pmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{pmatrix}
$$

The following pairs cannot be made orthogonal even though there is no product of permutation matrices with exactly one fixed point.

$(1,3),(1,9),(2,4),(2,7),(3,9),(4,7),(5,6),(5,8),(6,8),(10,11),(10,12),(11,12)$

The data above were obtained using

nofixedpoint := **proc**(*indlist1*, *indlist2*)
    local i, j, k, l, m, n;
        $i := \mathit{indlist1}_1$;
        $j := \mathit{indlist1}_2$;
        $k := \mathit{indlist1}_3$;
        $l := \mathit{indlist2}_1$;
        $m := \mathit{indlist2}_2$;
        $n := \mathit{indlist2}_3$;
        if not (trace(evalm($P.i$ ‘&∗‘ $P.l$)) = 1 **or** trace(evalm($P.i$ ‘&∗‘ $P.m$)) = 1 **or**
        trace(evalm($P.i$ ‘&∗‘ $P.n$)) = 1 **or** trace(evalm($P.j$ ‘&∗‘ $P.l$)) = 1 **or**
        trace(evalm($P.j$ ‘&∗‘ $P.m$)) = 1 **or** trace(evalm($P.j$ ‘&∗‘ $P.n$)) = 1 **or**
        trace(evalm($P.k$ ‘&∗‘ $P.l$)) = 1 **or** trace(evalm($P.k$ ‘&∗‘ $P.m$)) = 1 **or**
        trace(evalm($P.k$ ‘&∗‘ $P.n$)) = 1) **then** RETURN(*true*)
        else RETURN(*false*)
        fi
    end

and

```
two_ort_cartantest := proc(x1, x2)
  local eqs, U1, U2, p, i1, i2, k, q, p1, q1;
    eqs := {};
    U1 := cartanlist[x1];
    U2 := cartanlist[x2];
    for p to 3  do
      i1 := op(p, U1);
      i2 := op(p, U2);
      A1.p := evalm(diag(a1.p, b1.p, c1.p, d1.p, e1.p, f1.p) `&*` P.i1);
      A2.p := evalm(diag(a2.p, b2.p, c2.p, d2.p, e2.p, f2.p) `&*` P.i2)
    od;
    for k to 2  do
      for p to 3  do
        eqs := eqs union {a.k.p <> 0, b.k.p <> 0, c.k.p <> 0, d.k.p <> 0, e.k.p <> 0, f.k.p <> 0};
        A.k.1.p := submatrix(A.k.p, 1..3, 1..3);
        A.k.2.p := submatrix(A.k.p, 1..3, 4..6);
        A.k.3.p := submatrix(A.k.p, 4..6, 1..3);
        A.k.4.p := submatrix(A.k.p, 4..6, 4..6);
        T.k.1.p := evalm(A.k.1.p + transpose(A.k.4.p));
        T.k.2.p := evalm(A.k.2.p - transpose(A.k.2.p));
        T.k.3.p := evalm(A.k.3.p - transpose(A.k.3.p))
      od;
      for p to 3  dofor q to 3 do
          eqs := eqs union {trace(evalm(A1.p `&*` A2.q))};
          com.k.p.q := evalm((A.k.p `&*` A.k.q) - (A.k.q `&*` A.k.p));
          for p1 to 6 dofor q1 to 6 do
              eqs := eqs union {com.k.p.q[p1, q1] = 0};
              if p1 < 4 and q1 < 4 then eqs := eqs union {T.k.p.q[p1, q1] = 0} fi
            od
          od
        od
      od
    od;
    RETURN(solve(eqs))
  end
```

A.4 The orthogonal triples

In this list (i,j,k) refers to the triple (Hi,Hj,Hk) of subalgebras from the above list.

$(1, 15, 28)$ $(1, 15, 29)$ $(1, 16, 28)$ $(1, 16, 29)$ $(1, 19, 19)$ $(1, 19, 28)$
$(1, 19, 29)$

$(2, 17, 30)$ $(2, 17, 31)$ $(2, 18, 30)$ $(2, 18, 31)$ $(2, 19, 19)$ $(2, 19, 30)$
$(2, 19, 31)$

$(3, 19, 19)$ $(3, 19, 32)$ $(3, 19, 33)$ $(3, 20, 32)$ $(3, 20, 33)$ $(3, 21, 32)$
$(3, 21, 33)$

$(4,19,19)$ $(4,19,36)$ $(4,19,37)$ $(4,24,36)$ $(4,24,37)$ $(4,25,36)$
$(4,25,37)$

$(5,13,26)$ $(5,13,27)$ $(5,14,26)$ $(5,14,27)$ $(5,19,19)$ $(5,19,26)$
$(5,19,27)$

$(6,19,19)$ $(6,19,32)$ $(6,19,33)$ $(6,20,32)$ $(6,20,33)$ $(6,21,32)$
$(6,21,33)$

$(7,19,19)$ $(7,19,34)$ $(7,19,35)$ $(7,22,34)$ $(7,22,35)$ $(7,23,34)$
$(7,23,35)$

$(8,19,19)$ $(8,19,36)$ $(8,19,37)$ $(8,24,36)$ $(8,24,37)$ $(8,25,36)$
$(8,25,37)$

$(9,19,19)$ $(9,19,34)$ $(9,19,35)$ $(9,22,34)$ $(9,22,35)$ $(9,23,34)$
$(9,23,35)$

$(10,15,28)$ $(10,15,29)$ $(10,16,28)$ $(10,16,29)$ $(10,19,19)$ $(10,19,28)$
$(10,19,29)$

$(11,13,26)$ $(11,13,27)$ $(11,14,26)$ $(11,14,27)$ $(11,19,19)$ $(11,19,26)$
$(11,19,27)$

$(12,17,30)$ $(12,17,31)$ $(12,18,30)$ $(12,18,31)$ $(12,19,19)$ $(12,19,31)$
$(12,19,30)$

$(13,15,19)$ $(13,16,19)$ $(13,17,19)$ $(13,18,19)$ $(13,19,19)$ $(13,19,20)$
$(13,19,21)$ $(13,19,24)$ $(13,19,25)$ $(13,19,26)$ $(13,19,27)$ $(13,19,34)$
$(13,19,35)$ $(13,22,26)$ $(13,22,27)$ $(13,22,34)$ $(13,22,35)$ $(13,23,26)$
$(13,23,27)$ $(13,23,34)$ $(13,23,35)$

$(14,15,19)$ $(14,16,19)$ $(14,17,19)$ $(14,18,19)$ $(14,19,19)$ $(14,19,20)$
$(14,19,21)$ $(14,19,24)$ $(14,19,25)$ $(14,19,26)$ $(14,19,27)$ $(14,19,34)$
$(14,19,35)$ $(14,22,26)$ $(14,22,27)$ $(14,22,34)$ $(14,22,35)$ $(14,23,26)$
$(14,23,27)$ $(14,23,34)$ $(14,23,35)$

$(15,17,19)$ $(15,18,19)$ $(15,19,19)$ $(15,19,20)$ $(15,19,21)$ $(15,19,22)$
$(15,19,23)$ $(15,19,28)$ $(15,19,29)$ $(15,19,36)$ $(15,19,37)$ $(15,24,28)$
$(15,24,29)$ $(15,24,36)$ $(15,24,37)$ $(15,25,28)$ $(15,25,29)$ $(15,25,36)$
$(15,25,37)$

$(16,17,19)$ $(16,18,19)$ $(16,19,19)$ $(16,19,20)$ $(16,19,21)$ $(16,19,22)$
$(16,19,23)$ $(16,19,28)$ $(16,19,29)$ $(16,19,36)$ $(16,19,37)$ $(16,24,28)$
$(16,24,29)$ $(16,24,36)$ $(16,24,37)$ $(16,25,28)$ $(16,25,29)$ $(16,25,36)$
$(16,25,37)$

$(17,19,19)$ $(17,19,22)$ $(17,19,23)$ $(17,19,24)$ $(17,19,25)$ $(17,19,30)$
$(17,19,31)$ $(17,19,32)$ $(17,19,33)$ $(17,20,30)$ $(17,20,31)$ $(17,20,32)$
$(17,20,33)$ $(17,21,30)$ $(17,21,31)$ $(17,21,32)$ $(17,21,33)$

$(18,19,19)$ $(18,19,22)$ $(18,19,23)$ $(18,19,24)$ $(18,19,25)$ $(18,19,30)$
$(18,19,31)$ $(18,19,32)$ $(18,19,33)$ $(18,20,30)$ $(18,20,31)$ $(18,20,32)$
$(18,20,33)$ $(18,21,30)$ $(18,21,31)$ $(18,21,32)$ $(18,21,33)$

$(19,19,19)$ $(19,19,20)$ $(19,19,21)$ $(19,19,22)$ $(19,19,23)$ $(19,19,24)$
$(19,19,25)$ $(19,19,26)$ $(19,19,27)$ $(19,19,28)$ $(19,19,29)$ $(19,19,30)$
$(19,19,31)$ $(19,19,32)$ $(19,19,33)$ $(19,19,34)$ $(19,19,35)$ $(19,19,36)$
$(19,19,37)$ $(19,20,22)$ $(19,20,23)$ $(19,20,24)$ $(19,20,25)$ $(19,20,30)$
$(19,20,31)$ $(19,20,32)$ $(19,20,33)$ $(19,21,22)$ $(19,21,23)$ $(19,21,24)$
$(19,21,25)$ $(19,21,30)$ $(19,21,31)$ $(19,21,32)$ $(19,21,33)$ $(19,22,24)$
$(19,22,25)$ $(19,22,26)$ $(19,22,27)$ $(19,22,34)$ $(19,22,35)$ $(19,23,24)$
$(19,23,25)$ $(19,23,26)$ $(19,23,27)$ $(19,23,34)$ $(19,23,35)$ $(19,24,28)$
$(19,24,29)$ $(19,24,36)$ $(19,24,37)$ $(19,25,28)$ $(19,25,29)$ $(19,25,36)$
$(19,25,37)$ $(19,26,28)$ $(19,26,29)$ $(19,26,30)$ $(19,26,31)$ $(19,26,32)$
$(19,26,33)$ $(19,26,34)$ $(19,26,35)$ $(19,26,36)$ $(19,26,37)$ $(19,27,28)$
$(19,27,29)$ $(19,27,30)$ $(19,27,31)$ $(19,27,32)$ $(19,27,33)$ $(19,27,34)$
$(19,27,35)$ $(19,27,36)$ $(19,27,37)$ $(19,28,28)$ $(19,28,29)$ $(19,28,30)$
$(19,28,31)$ $(19,28,32)$ $(19,28,33)$ $(19,28,34)$ $(19,28,35)$ $(19,28,36)$
$(19,28,37)$ $(19,29,29)$ $(19,29,30)$ $(19,29,31)$ $(19,29,32)$ $(19,29,33)$
$(19,29,34)$ $(19,29,35)$ $(19,29,36)$ $(19,29,37)$ $(19,30,30)$ $(19,30,31)$
$(19,30,32)$ $(19,30,33)$ $(19,30,34)$ $(19,30,35)$ $(19,30,36)$ $(19,30,37)$
$(19,31,31)$ $(19,31,32)$ $(19,31,33)$ $(19,31,34)$ $(19,31,35)$ $(19,31,36)$
$(19,31,37)$ $(19,32,32)$ $(19,32,33)$ $(19,32,34)$ $(19,32,35)$ $(19,32,36)$

$(19,32,37)$ $(19,33,33)$ $(19,33,34)$ $(19,33,35)$ $(19,33,36)$ $(19,33,37)$
$(19,34,34)$ $(19,34,35)$ $(19,34,36)$ $(19,34,37)$ $(19,35,35)$ $(19,35,36)$
$(19,35,37)$ $(19,36,36)$ $(19,36,37)$ $(19,37,37)$

$(26,34,34)$ $(26,34,35)$ $(26,35,35)$

$(27,34,34)$ $(27,34,35)$ $(27,35,35)$

$(28,28,36)$ $(28,28,37)$ $(28,29,36)$ $(28,29,37)$ $(28,36,36)$ $(28,36,37)$
$(28,37,37)$

$(29,29,36)$ $(29,29,37)$ $(29,36,36)$ $(29,36,37)$ $(29,37,37)$

$(30,30,32)$ $(30,30,33)$ $(30,31,32)$ $(30,31,33)$ $(30,32,32)$ $(30,32,33)$
$(30,33,33)$

$(31,31,32)$ $(31,31,33)$ $(31,32,32)$ $(31,32,33)$ $(31,33,33)$

The triples were calculated with the following procedure:

```
three_ort_cartantest := proc(x1, x2, x3)
  local U1, U2, U3, eqs, p, i1, i2, i3, k, q, y, z, p1, q1;
    U1 := cartanlist[x1];
    U2 := cartanlist[x2];
    U3 := cartanlist[x3];
    eqs := {};
    for p to 3 do
      i1 := op(p, U1);
      i2 := op(p, U2);
      i3 := op(p, U3);
      A1.p := evalm(diag(a1.p, b1.p, c1.p, d1.p, e1.p, f1.p) `&*` P.i1);
      A2.p := evalm(diag(a2.p, b2.p, c2.p, d2.p, e2.p, f2.p) `&*` P.i2);
      A3.p := evalm(diag(a3.p, b3.p, c3.p, d3.p, e3.p, f3.p) `&*` P.i3)
    od;
    for k to 3 do
      for p to 3 do
        eqs := eqs union {c.k.p <> 0, b.k.p <> 0, a.k.p <> 0, d.k.p <> 0, e.k.p <> 0, f.k.p <> 0};
        A.k.1.p := submatrix(A.k.p, 1..3, 1..3);
        A.k.2.p := submatrix(A.k.p, 1..3, 4..6);
        A.k.3.p := submatrix(A.k.p, 4..6, 1..3);
        A.k.4.p := submatrix(A.k.p, 4..6, 4..6);
        T.k.1.p := evalm(A.k.1.p + transpose(A.k.4.p));
        T.k.2.p := evalm(A.k.2.p - transpose(A.k.2.p));
        T.k.3.p := evalm(A.k.3.p - transpose(A.k.3.p))
      od;
      for p to 3 do for q to 3 do
          for y to 3 do for z to 3 do
              if not (y = z) then eqs := eqs union {trace(evalm(A.y.p `&*` A.z.q))} fi
            od
          od;
          com.k.p.q := evalm((A.k.p `&*` A.k.q) - (A.k.q `&*` A.k.p));
          for p1 to 6 do for q1 to 6 do
              eqs := eqs union {com.k.p.q[p1, q1] = 0};
              if p1 < 4 and q1 < 4 then eqs := eqs union {T.k.p.q[p1, q1] = 0} fi
            od
          od
        od
      od
    od;
    RETURN(solve(eqs))
```

A.5 Possible six tuples of subalgebras

This is a list of all six tuples of subalgebras such that any three of can be made orthogonal.

$$
\begin{array}{lll}
(19,19,26,26,34,34) & (19,19,26,26,34,35) & (19,19,26,26,35,35) \\
(19,19,26,27,34,34) & (19,19,26,27,34,35) & (19,19,26,27,35,35) \\
(19,19,27,27,34,34) & (19,19,27,27,34,35) & (19,19,27,27,35,35) \\
(19,19,28,28,36,36) & (19,19,28,28,36,37) & (19,19,28,28,37,37) \\
(19,19,28,29,36,36) & (19,19,28,29,36,37) & (19,19,28,29,37,37) \\
(19,19,29,29,36,36) & (19,19,29,29,36,37) & (19,19,29,29,37,37) \\
(19,19,30,30,32,32) & (19,19,30,30,32,33) & (19,19,30,30,33,33) \\
(19,19,30,31,32,32) & (19,19,30,31,32,33) & (19,19,30,31,33,33) \\
(19,19,31,31,32,32) & (19,19,31,31,32,33) & (19,19,31,31,33,33)
\end{array}
$$

In order to minimize the total amount of calculations all five tuples of orthogonal subalgebras were calculated first:

fivelist := **proc**(*triplelist*)

local *fivelist*, *triple*, *i*, *j*, *fivetuple*, *fivetuple_od*, *perm*;

fivelist := [] ;

for *triple* **in** *triplelist* **dofor** *i* **to** 37 **dofor** *j* **to** 37 **do**

fivetuple := sort([op(*triple*), *i*, *j*]) ;

fivetuple_od := *true* ;

for *perm* **in** choose([1, 2, 3, 4, 5], 3) **do**

if not

member($[fivetuple_{perm_1}, fivetuple_{perm_2}, fivetuple_{perm_3}]$, *triplelist*)

then *fivetuple_od* := *false*

fi

od;

if *fivetuple_od* **and not** member(*fivetuple*, *fivelist*) **then**

print(*triple*, *fivetuple*) ; *fivelist* := [op(*fivelist*), *fivetuple*]

fi

od

od

od;

RETURN(*fivelist*)

end

and then all completions to six tuples

```
sixlist := proc(fivelist, triplelist)
    local sixlist, fivetuple, i, sixtuple, sixtuple_od, perm;
      sixlist := [] ;
      for fivetuple  in fivelist dofor i to  37 do
            sixtuple := sort([op(fivetuple), i]) ;
            sixtuple_od := true ;
            for perm in choose([op(fivetuple)], 2)  do
               if  not  member(sort([op(perm), i]), triplelist) then sixtuple_od := false fi
            od;
            if sixtuple_od and   not member(sixtuple, sixlist) then
               print(fivetuple, sixtuple) ; sixlist := [op(sixlist), sixtuple]
            fi
         od
      od;
      RETURN(sixlist)
    end
```

Finally those with more than 2 occurrences of $H19$ were removed resulting in the 18 six tuples above which were all examined by

```
six_ort_cartantest := proc(x1, x2, x3, x4, x5, x6)
  local U1, U2, U3, U4, U5, U6, eqs, p, i1, i2, i3, i4, i5, i6, k, q, y, z, p1, q1;
    U1 := cartanlist[x1] ;
    U2 := cartanlist[x2] ;
    U3 := cartanlist[x3] ;
    U4 := cartanlist[x4] ;
    U5 := cartanlist[x5] ;
    U6 := cartanlist[x6] ;
    eqs := {} ;
    for p to 3 do
      i1 := op(p, U1) ;
      i2 := op(p, U2) ;
      i3 := op(p, U3) ;
      i4 := op(p, U4) ;
      i5 := op(p, U5) ;
      i6 := op(p, U6) ;
      A1.p := evalm(diag(a1.p, b1.p, c1.p, d1.p, e1.p, f1.p) `&*` P.i1) ;
      print(A1.p) ;
      A2.p := evalm(diag(a2.p, b2.p, c2.p, d2.p, e2.p, f2.p) `&*` P.i2) ;
      print(A2.p) ;
      A3.p := evalm(diag(a3.p, b3.p, c3.p, d3.p, e3.p, f3.p) `&*` P.i3) ;
      print(A3.p) ;
      A4.p := evalm(diag(a4.p, b4.p, c4.p, d4.p, e4.p, f4.p) `&*` P.i4) ;
      print(A4.p) ;
      A5.p := evalm(diag(a5.p, b5.p, c5.p, d5.p, e5.p, f5.p) `&*` P.i5) ;
      print(A5.p) ;
      A6.p := evalm(diag(a6.p, b6.p, c6.p, d6.p, e6.p, f6.p) `&*` P.i6) ;
      print(A6.p)
    od;
    for k to 6 do
      for p to 3 do
        eqs := eqs union {d.k.p ≠ 0, e.k.p ≠ 0, f.k.p ≠ 0, c.k.p ≠ 0, a.k.p ≠ 0, b.k.p ≠ 0} ;
        A.k.1.p := submatrix(A.k.p, 1..3, 1..3) ;
        A.k.2.p := submatrix(A.k.p, 1..3, 4..6) ;
        A.k.3.p := submatrix(A.k.p, 4..6, 1..3) ;
        A.k.4.p := submatrix(A.k.p, 4..6, 4..6) ;
        T.k.1.p := evalm(A.k.1.p + transpose(A.k.4.p)) ;
        T.k.2.p := evalm(A.k.2.p − transpose(A.k.2.p)) ;
        T.k.3.p := evalm(A.k.3.p − transpose(A.k.3.p))
      od;
```